**校园社区平台**

作者姓名：赵安玉

专业名称：软件工程

指导教师：张玺 助教

**成都理工大学工程技术学院**

**学位论文诚信承诺书**

**本人慎重承诺和声明：**

1.本人已认真学习《学位论文作假行为处理办法》（中华人民共和国教育部第34号令）、《成都理工大学工程技术学院学位论文作假行为处理实施细则（试行）》（成理工教发〔2013〕30号）文件并已知晓教育部、学院对论文作假行为处理的有关规定，知晓论文作假可能导致作假者被取消学位申请资格、注销学位证书、开除学籍甚至被追究法律责任等后果。

2.本人已认真学习《成都理工大学工程技术学院毕业设计指导手册》，已知晓学院对论文撰写的内容和格式要求。

3.本人所提交的学位论文（题目：网上拍卖系统），是在指导教师指导下独立完成，本人对该论文的真实性、原创性负责。若论文按有关程序调查后被认定存在作假行为，本人自行承担相应的后果。

承诺人（学生签名）：

20 年 月 日

注：学位论文指向我校申请学士学位所提交的本科学生毕

**摘要**

随着信息技术的发展，以及移动终端设备普及，互联网融入我们日常生活中，信息交流、信息共享无时无刻不出现在我们身边，因此网络上发布了各种社交软件、社交平台，然而这些平台很少有面向的大学生这用户群体。所以校园社交平台的研究与开发很重要，有利于高校对自己学校平台用户的管理，无论学生还是老师，只要是本校的用户都可以通过本校平台去发布或寻找有效的信息。

校园社区平台，在实际生活主要用于高校自己内部职员或学生，发布或者搜寻信息。平台的管理人员由高校自己分配，对平台信息审核也有管理自己人员进行，这样有利于信息的真实性。平台的用户可以是学生，也可以是老师，或者学校在职人员，在一定程度上限制了用户群体，有保证了信息的范围性；用户在平台上可以查询信息，也可以发布信息，信息可以是多种类型视频、文本、图片等，这样可以充分保证内容的多样性，比如课程，日常VLOG,寻物启事等等。但是用户发布内容需要遵循平台的规则，平台对敏感字进行过滤。

本平台设计实现技术采用SpringBoot作为后台的主体框架，整合MyBtisPlus作为持久层框架，连接MySQL数据库,对数据库中的表进行增加、查询、修改、删除等操作。使用Redis来实现平台的数据缓存，对一些热点的信息进行缓存，这样避免了数据库写的压力过大，除此还对激活码这类临时的一次性信息进行缓存。与前端交互的API则由GraphQL实现 项目构建，精确的查询结果展示和单一的URL接口，不仅节省了网络资源，而且避免了大量的URL接口。信息的查询遵循GraphQL官方的语法实现。前端选用Vue作为前端框架，页面的实现使用了ElementUI搭建，API交互使用GraphQL vue-apollo项目构建。最后对平台各个功能模块进行测试，根据测试证实了，各个模块所实现的功能，基本满足了平台的日常使用，整个设计达到预期设计目标。

**关键词**：校园社区平台 SpringBoot GraphQL Redis Vue

**Abstract**

目录

[摘要 I](#_Toc1025202362)

[Abstract 2](#_Toc1350490027)

[前言 - 1 -](#_Toc783368690)

[1 绪论 - 2 -](#_Toc1102520059)

[2 校园社区平台技术概述 - 3 -](#_Toc2044897763)

[2.1 概念 - 3 -](#_Toc1967513926)

[2.1.1 基本概念 - 3 -](#_Toc1365180540)

[3 平台架构设计 - 4 -](#_Toc1540383426)

[3.1 开发技术及版本依赖 - 4 -](#_Toc304089172)

[3.1.1 开发环境 - 4 -](#_Toc1303455736)

[3.1.2 开发技术 - 4 -](#_Toc35005211)

[3.2 平台架构 - 4 -](#_Toc521595368)

[3.2.1 平台后端模块概述 - 4 -](#_Toc294702567)

[3.2.2 平台前端概述 - 5 -](#_Toc1726956429)

[3.2.3 平台前后端通信技术 - 5 -](#_Toc336465782)

[4 平台详细设计及实现 - 6 -](#_Toc861021530)

[4.1 平台数据存储设计 - 6 -](#_Toc278722862)

[4.1.1 MySQL数据库表设计 - 6 -](#_Toc233665123)

[4.2 平台后端设计与实现 - 9 -](#_Toc2145174067)

[4.2.1 功能模块设计及概要 - 9 -](#_Toc468703135)

[4.2.2 部分功能设计与实现 - 20 -](#_Toc1101513929)

[4.3 平台前端设计与实现 - 20 -](#_Toc1801979802)

[4.3.1 前端技术选取分析 - 20 -](#_Toc1315634022)

[4.3.2 页面设计与布局 - 20 -](#_Toc635723058)

[4.4 平台前后数据交互策略及实现 - 20 -](#_Toc1369133069)

[4.4.1 交互技术选取技术分析 - 20 -](#_Toc1125898167)

[4.4.2 数据通信详细设计 - 20 -](#_Toc1059961393)

[4.5.3 数据交互详细实现 - 20 -](#_Toc2089018456)

[5 测试与分析 - 21 -](#_Toc628175011)

[5.1 测工具介绍 - 21 -](#_Toc1656478042)

[5.2 测试技术与方法概述 - 21 -](#_Toc1131176229)

[5.2 测试用例 - 21 -](#_Toc1653377373)

[总结 - 22 -](#_Toc859484421)

[参考文献 - 23 -](#_Toc1914544919)

**前言**

随着互联网的不断发展，人们获取信息的途径也多样化，但是伴随着的问题也越来越多。对于高校学生和职员，获取或发布信息多数借助于第三方平台，会带来的一系列问题，首先，平台上发布信息的真实性不能确定；其次，信息的时效性不能保证，不能保证信息及时和有效；然后，信息的可控性，学校或者个人难以通过第三方平台控制不实信息的传播。因此，推动高校自身的社区平台发展势在必行。如今，开发后端语言Java作为生态最好，外加Springt体系的发展，选取它作为后端开发技术;前端目前Vue框架是国内使用最多、最流行开源框架。所以，我会将这些技术应用于本次毕业设计中。

现在无论是开发技术，还是开发工具互联网都有大量的资源，本次毕设选取对Java变成支持最好的开发工具IDEA 来进行开发。本文参照了部分开源软件设计和现有的视频教学，结合大学所学习的专业知识，以及市面上社交软件的功能，设计并开发了一款前后端分离的校园社区平台的Web应用。此外没害使用了Docker，数据库管理工具，测试工具等等对系统进行拓展与开发，来达到平台设计目标。

借助该平台，学校管理人员可以控制信息的有效性、真实性，以及注册用户来源的详细信息；本校也可以使用官方账号来发布官方信息，这样极大的保证了平台用户体验友好性，用户获可以不用从其他平台信息获取信息，平台用户可以发布和浏览信息。

**1 绪论**

如今，随着在线消费成为主流趋势，淘宝和京东等在线购物平台已广为人知，而在线购物是公众购买商品和公司出售商品的最佳方式轻松，方便，快捷。杂货，服装，电子产品，奇怪的小商品，日用品等都是各种商品。无需离开房屋，即可通过在线平台轻松搜索和购买一些按钮。以合理的价格，便利的购物和上门服务，所有这些都是在线购物的法宝，可以在当今竞争激烈的市场环境中赢得许多消费者。特别是在当今快节奏和

# 2 校园社区平台技术概述

**2.1 概念**

### 2.1.1 基本概念

1、J2EE的概念

J2EE技术是一种通过使用了Java的平台和数据库来有效简化与企业和公司系统解决方案的设计和开发,增长和有效管理与业务有关的复杂问题的系统结构。

2、模块化设计的概念和思想

模块化[程序设计](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1/223952" \t "/home/anyu/Documents\\x/_blank)是指在进行[程序设计](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1" \t "/home/anyu/Documents\x/_blank)时将一个大程序按照功能划分为若干小程序模块，每个小程序模块完成一个确定的功能，并在这些模块之间建立必要的联系，通过模块的互相协作完成整个功能的程序设计方法。在设计较复杂的程序时，一般采用自顶向下的方法，将问题划分为几个部分，各个部分再进行细化，直到分解为较好解决问题为止。模块化的目的是为了降低程序复杂度，使程序设计、调试和维护等操作简单化。

前后端分离架构技术概念

3、MVC 设计模式概念

4、Spring 框架

Spring框架是[Java平台](https://baike.baidu.com/item/Java%E5%B9%B3%E5%8F%B0/3793459" \t "/home/anyu/Documents\\x/_blank)上的一种开源应用框架，提供具有控制反转特性的容器。尽管Spring框架自身对编程模型没有限制，但其在Java应用中的频繁使用让它备受青睐，以至于后来让它作为[EJB](https://baike.baidu.com/item/EJB/144195" \t "/home/anyu/Documents\\x/_blank)（EnterpriseJavaBeans）模型的补充，甚至是替补。它简化了 Java 企业级应用开发。

5、SpingBoot 概念

6、MybatisPlus 框架概念

7、GraphQL API 查询语言的概念

8、JWT (Java Web Token)的概念

# 3 平台架构设计

**3.1 开发技术及版本依赖**

本节主主要介绍校园社区平台所用到的技术和开发环境，以及所依赖其他资源。

### 3.1.1 开发环境

操作系统：Linux 510

开发工具：IDEA 2020.3 Navicat

数据库：MySQL 8.0.20 Redis

容器技术：Docker

测试工具：Postman

SpringBoot 2.3.4.RELEASE

MybatisPlus 3.4.0

Java 11

Gradle 6.7

Vue 2

### 3.1.2 开发技术

**3.2 平台架构**

平台是一个前后端分离的web应用。后端使用了Gradle作为版本控制工具和模块管理工具；基于SpringBoot 的框架做快速开发; MybatisPlus做持久层框架,进行数据库操作。前端使用Vue框架搭建前端应用，利用Element 进行快速页面搭建。使用GraphQL作为前后端通信方式。

### 3.2.1 平台后端模块概述

各个模块遵循软件开发模块化设计原则，每个模块都只关注自己的功能或业务。其中ancf-msg、ancf-user、ancf-post业务模块按MVC模式实现，它们是和前端进行交互的模块；理论上 ancf-auth、ancf-oss、ancf-cache对外部模块只暴露业务接口，它们只负责对内部模块提供服务。

1、 ancf-webapp 模块，它是程序的主入口模块，最顶层模块；直接依赖于ancf-service-msg、ancf-service-user、ancf-service-post业务模块。程序的配置文件存放该模块中。

2、ancf-service-msg 模块，它是校园社区平台用户间消息发送、平台消息推送的业务实现。

3、 ancf-service-user 模块，它是校园社区平台与用户相关业务的实现，包括用户注册、用户登录、头像修改等等。

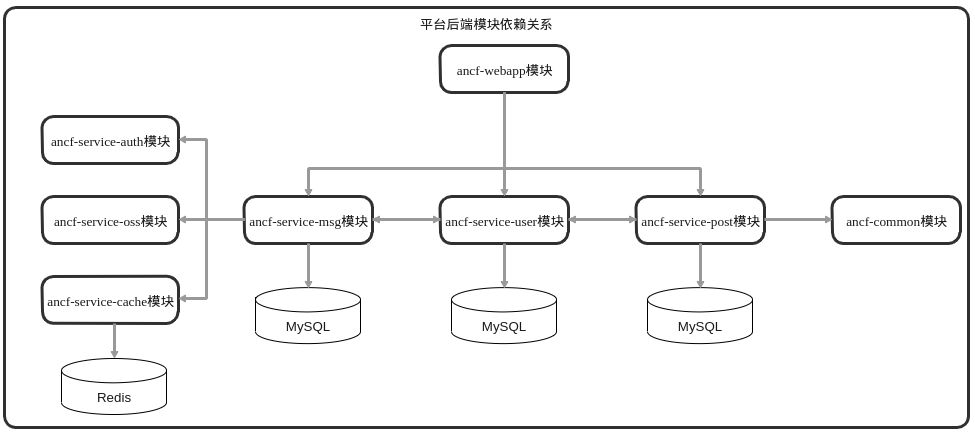
4、 ancf-service-post 模块，它是校园社区论坛与帖子相关业务的实现，包括发帖，评论、点赞等等。

5、ancf-common 模块，它是公共模块，最底层模块，基本上所有的模块都依赖于它。

6、ancf-service-auth 模块，它是平台权限认证、安全管理的实现。

7、 ancf-service-oss模块，它是平台文件上传云端服务器的实现，借助于阿里云OSS服务。

8、 ancf-sesrvice-cache 模块，它是系统缓存的业务实现。



### 3.2.2 平台前端概述

### 3.2.3 平台前后端通信技术

# 4 平台详细设计及实现

本届主要介绍平台数据库、后端架构和前端架构的详细设计。

**4.1 平台数据存储设计**

关系型数据库使用MySQL，它作为平台主要的数据存储数据库，小部分实时热点数据或者有效期的数据使用Redis作为缓存数据库。

### 4.1.1 MySQL数据库表设计

对于实体表，提供了通用的字段表示必要信息。例如create\_time 创建时间、modified\_time修改时间、status状态信息。

1. 用户表 user，用户表包含平台用户的个人信息和认证信息以及一些账号状态信息。账号、id和邮箱都是唯一的，可用作登录。头像存储的是阿里云对象存储服务的图片地址。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | ID |
| account | char | 16 | NOT NULL | 账号 |
| gender | tinyint | 1 | NOT NULL | 性别 |
| age | tinyint | 3 | NOT NULL | 年龄 |
| birthday | datetime | - | NOT NULL | 出生日期 |
| nickname | char | 9 | NOT NULL | 昵称 |
| real\_name | char | 6 | NULL | 真实姓名 |
| avatar | char | 100 | NOT NULL | 头像存储地址 |
| email | char | 25 | NULL | 邮箱地址 |
| mobile | char | 11 | NULL | 手机号 |
| salt | char | 5 | NOT NULL | 密码加盐 |
| password | char | 50 | NOT NULL | 密码 |
| activation | tinyint | 1 | NOT NULL | 账号是否激活 |
| status | tinyint | 1 | NOT NULL | 用户状态 |
| create\_time | datetime | - | NOT NULL | 用户创建时间 |
| modified\_time | datetime | - | NOT NULL | 用户修改时间 |

1. 评论表 comment

评论表用于帖子的评论、评论的评论、或者评论的回复的表示，帖子的评论的字段entity\_type 必须为 1,target\_id 为null;评论的评论的字段entity\_type 必须为2，target\_id为null，评论的回复entity\_type 为3，target\_id 为null。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | 评论id |
| content | varchar | 255 | NOT NULL | 内容 |
| user\_id | int | 10 | NOT NULL | 评论者id |
| entity\_id | int | 10 | NOT NULL | 评论的实体id |
| target\_id | int | 10 | NULL | 被评论者的id |
| entity\_type | tinyint | 1 | NOT NULL | 评论实体类型 |
| status | tinyint | 1 | NOT NULL | 评论状态 |
| create\_time | datetime | - | NOT NULL | 评论创建时间 |
| modifid\_time | datetime | - | NOT NULL | 评论修改时间 |

1. 帖子表 post

帖子的相关信息，

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | 帖子id |
| type | tinyint | 1 | NOT NULL | 帖子类型 |
| user\_id | int | 10 | NOT NULL | 发帖者id |
| title | varchar | 20 | NOT NULL | 帖子标题 |
| content | text | - | NOT NULL | 内容 |
| cmt\_num | int | 10 | NOT NULL | 评论数 |
| score | float | - | NOT NULL | 得分 |
| status | tinyint | 1 | NOT NULL | 帖子状态 |
| create\_time | datetime | - | NOT NULL | 创建时间 |
| modified\_time | datetime | - | NOT NULL | 修改时间 |

1. 消息表 message

私信或者平台内部消息的实体、

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | 消息id |
| from\_id | int | 10 | NOT NULL | 发送者id |
| to\_id | int | 10 | NOT NULL | 接受者id |
| chart\_id | varchar | 20 | NOT NULL | 聊天id |
| content | varchar | 255 | NOT NULL | 内容 |
| status | tinyint | 1 | NOT NULL | 状态 |
| create\_time | datetime | - | NOT NULL | 创建时间 |
| modified\_time | datetime | - | NOT NULL | 修改时间 |

**4.1.2 Redis缓存设计**

1、 设计缘由以及问题分析

首先，考虑到部分数据修改和查询频繁，直接访问MySQL数据库会造成其压力过大，从而导致程序效率问题。其次，像激活码这种临时的一次性数据，这类数据具有时效性，过期的数据没有任何可利用价值。如果存放到MySQL数据库中，仅仅只会被使用一次而且没有重复利用的价值，显然不合理。因此，我为平台设计数据缓存服务。

1. 设计方案与实现

借鉴目前缓存设计的方案，我在本地缓存、Redis以及Memcached

三者之间进行对比选择。首先，三者都是以Key-Value的形式存储，本地缓存适用于数据量较少且对数据操纵简单的数据，Redis可以支持更复杂的操作，如设置过期时间；所以本地缓存方案被否定。然后，Redis支持String、Hash、Set 、List和Sorted Set等五种数据操作，而Memcached只支持String，并且Redis支持数据持久化，而Memcached不支持。综上所述，从技术和实现功能难易程度两个方面考虑，选择Redis作为平台的缓存数据库是最合理方案。

**4.2 平台后端设计与实现**

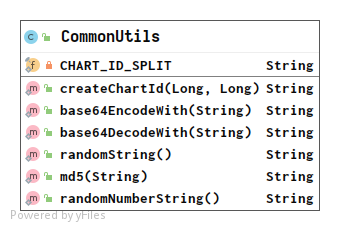
该章节介绍后端平台架构，各个模块实现的功能和所使用技术。以及它们在平台具有的作用。

### 4.2.1 功能模块设计及概要

**1、ancf-common**

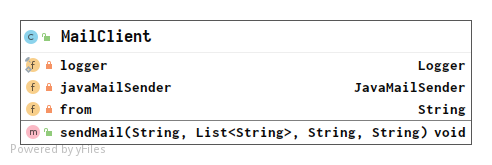
1. CommonUtils 平台公共工具类

提供给其他模块的公用工具类，里面包括了base64加密解密、生成随机字符串、生成随机数字、md5加密。

****

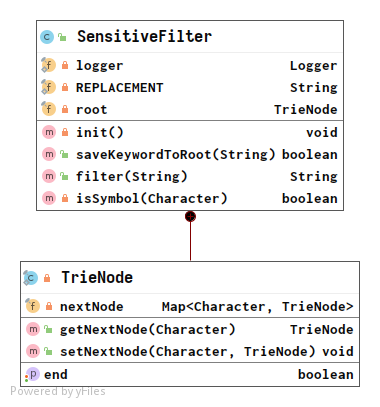
1. MailClient 系统邮件发送组件类

邮件客户端使用了springboot自带的邮件依赖，对其进行了自定义封装，通过调用sendMail方法发送邮件，可以一次分发多条邮件信息。

****

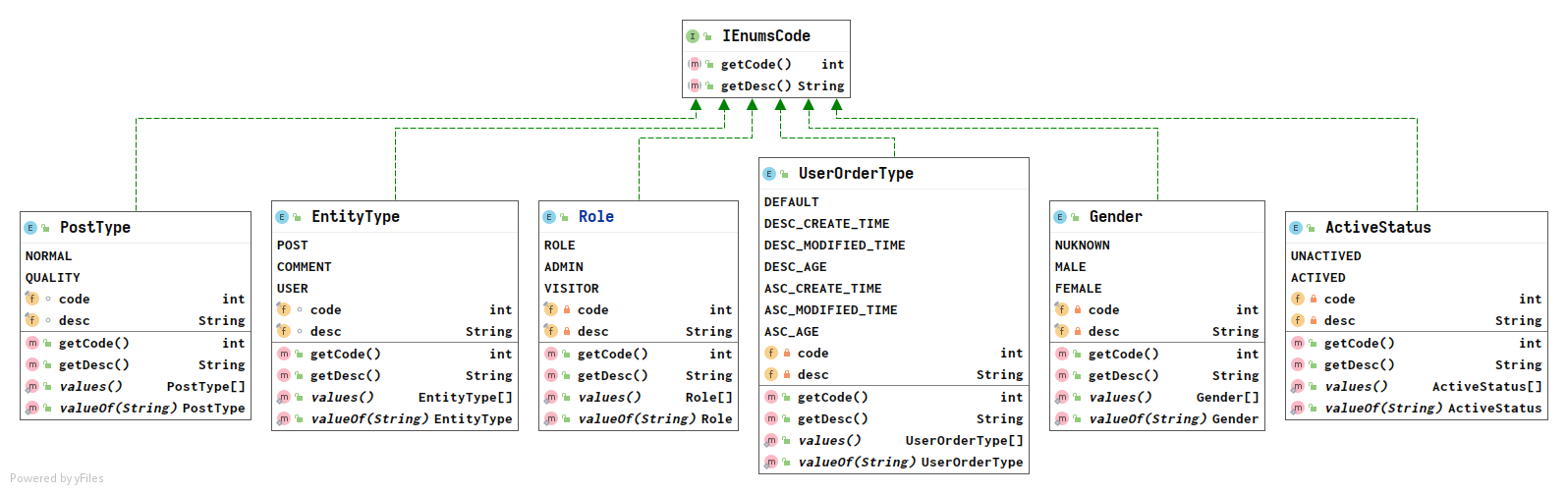
1. 平台敏感字文本文件过滤类

对于一个一个公共的平台，需要对敏感信息进行过滤处理，SensitiveFilter 正是平台的敏感词过滤处理工具类。其主要实现借助于字典树这一数据结构。init方法在初始化时会读取平台的敏感字文件，将每一个单词通过saveKeywordToRoot 方法加入到敏感字字典树中。最后生成一个有所有敏感字信息的字典树。filter方法对传入的文本文件结合已生成的敏感字字典树进行算法处理，最后返回过滤的文本。

****

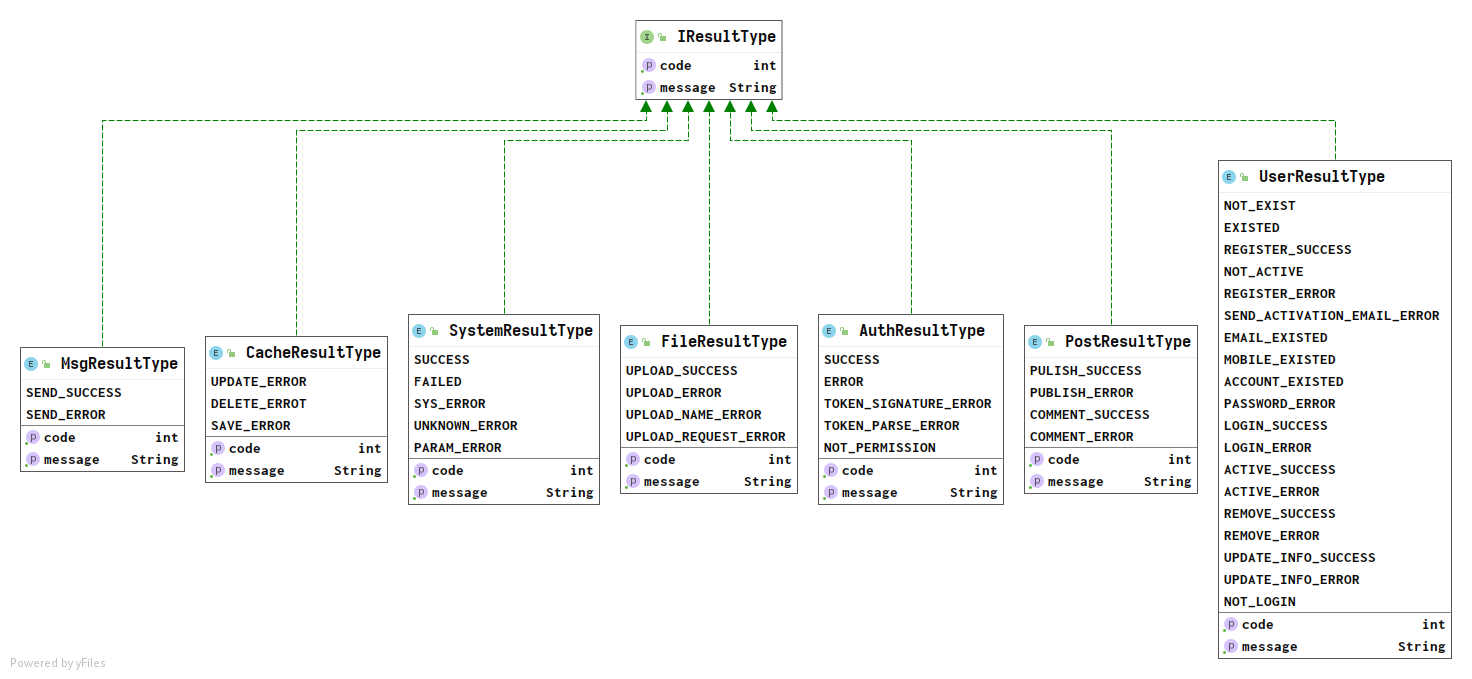
1. 数据库字典枚举类实现

数据库的某一些字段表示的值或者内容范围较小且固定，在设计数据库这些字段时，考虑到性能与合理性，通常字段类型使用tinyint类型；与Java中的Integer相互对应，考虑到数字所表示的信息在实际代码开发中对开发人员不够清晰，因此使用枚举俩定义这些字段；本平台的数据库字典枚举都继承IEmnuCode接口，通过实现该接口用来表示这是数据库字典，getCode 方法方法获取该枚举所存入数据库的值，getDesc 方法获取该枚举的描述信息。

****

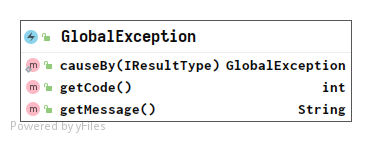
1. 业务处理结果状态枚举类

考虑到任何有意义的请求，服务器端都会进行业务操作，最终得到有且只有一个结果状态，返回给客户端。因此平台设计对此类结果状态进行统一化，规范化。所有业务处理结果的状态的枚举都实现IResultType接口，该接口有且只有三个方法，分别是isSuccess方法、 getCode方法 和getMessage方法；isSuccess方法表示业务处理结果是否成功，getCode方法返回业务处理结果的业务代码，getMessage方法返回业务处理结果的业务结果信息。

****

1. 自定义全局异常类

这是本平台所自定义的异常类继承于RunTimeException异常类，添加了code属性，重写了getMessage方法，返回业务结果状态信息。提供了一个静态 causeBy方法，用于通过业务处理结果状态，生成异常信息。

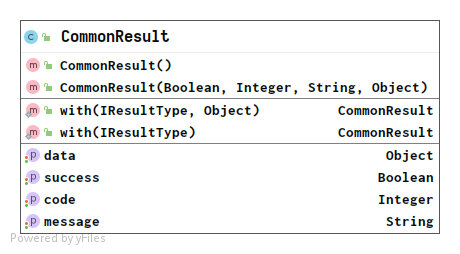


1. 全局异常处理，实现GraphQLErrorHandler接口，重写了processErrors方法。只返回给浏览器业务处理错误信息和业务代码。



1. 统一结果返回

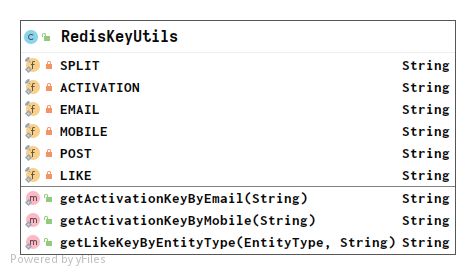
用于mutation类型的数据结果返回给客户端。提供了静态的with方法，可通过传入业务处理结果状态枚举类，来生成统一结果。



2、ancf-service-cache模块

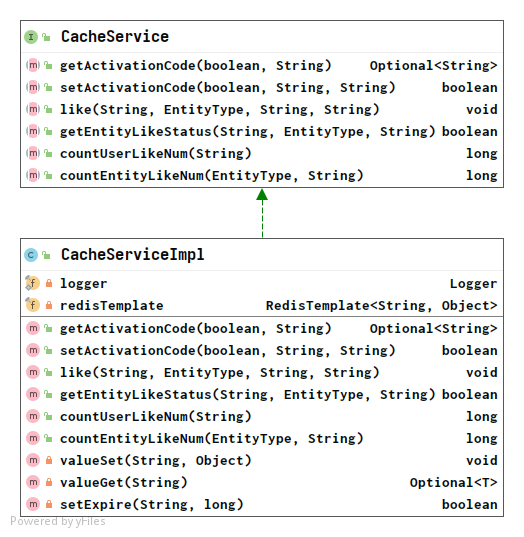
1. Redis 键生成工具类

缓存信息时，需要根据某些信息生成特定的键，方便再次从缓存中获取缓存信息。



1. 缓存服务接口及实现

其他模块使用缓存是只能通过调用缓存接口，进行缓存功能的实现。目前系统对账号激活的激活码、帖子的点赞数和点赞状态进行缓存。

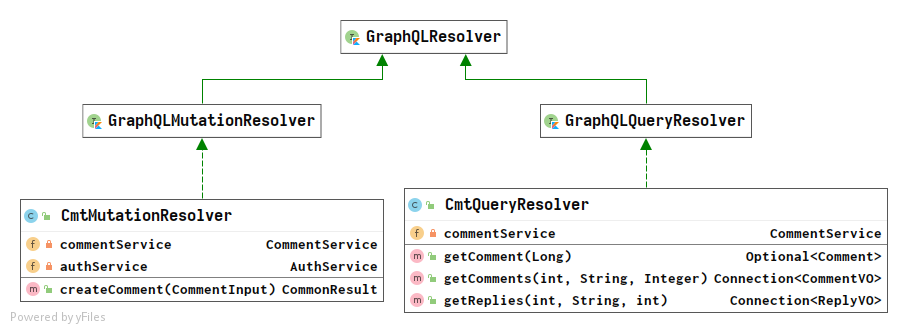
****

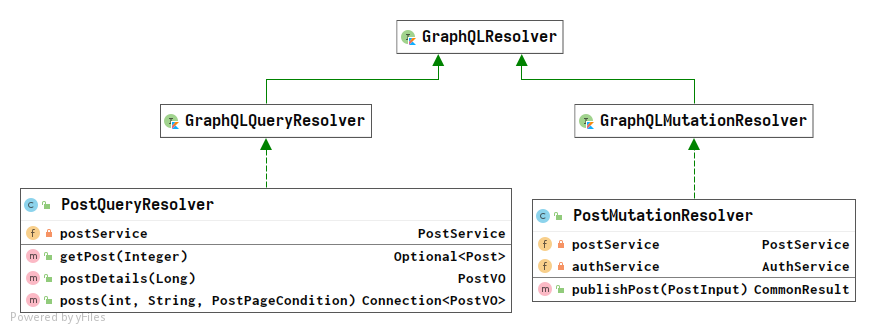
3、ancf-service-msg 模块

4、ancf-service-post 模块

1. 帖子API数据交互层,

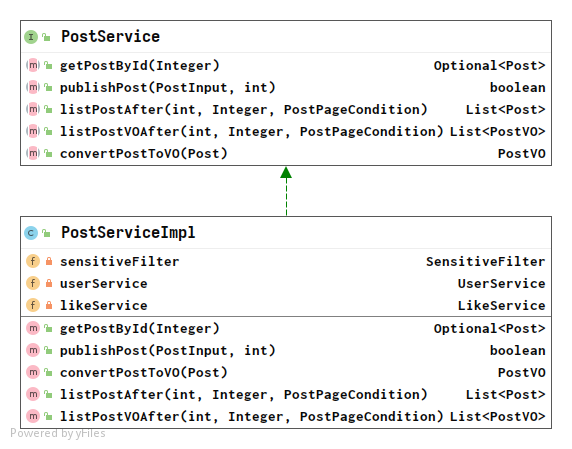
实现GraphQLMutationResolver接口,为数据变更操作;实现GraphQueryResolver接口，为数据查询操作。帖子评论功能模块，有创建评论和分页查询评论功能。

****

****

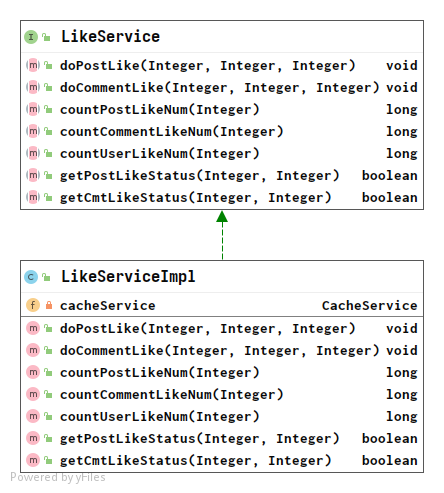
1. 帖子服务层，业务实现

查询根据帖子id帖子，分页查询帖子列表，发布帖子，帖子实体类类型装换等功能实现。

****

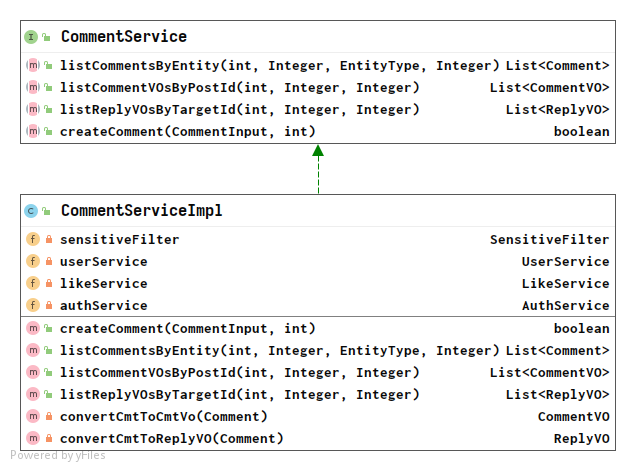
1. 点赞业务层

帖子和评论点赞的业务逻辑实现，查询帖子或评论的点赞数、点赞状态，点赞以及取消点赞。



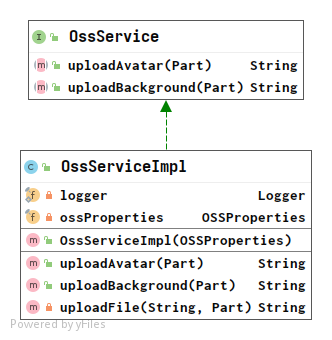
1. 评论业务层实现

根据不同的实体查询评论,创建评论，分页查询评论列表。

****

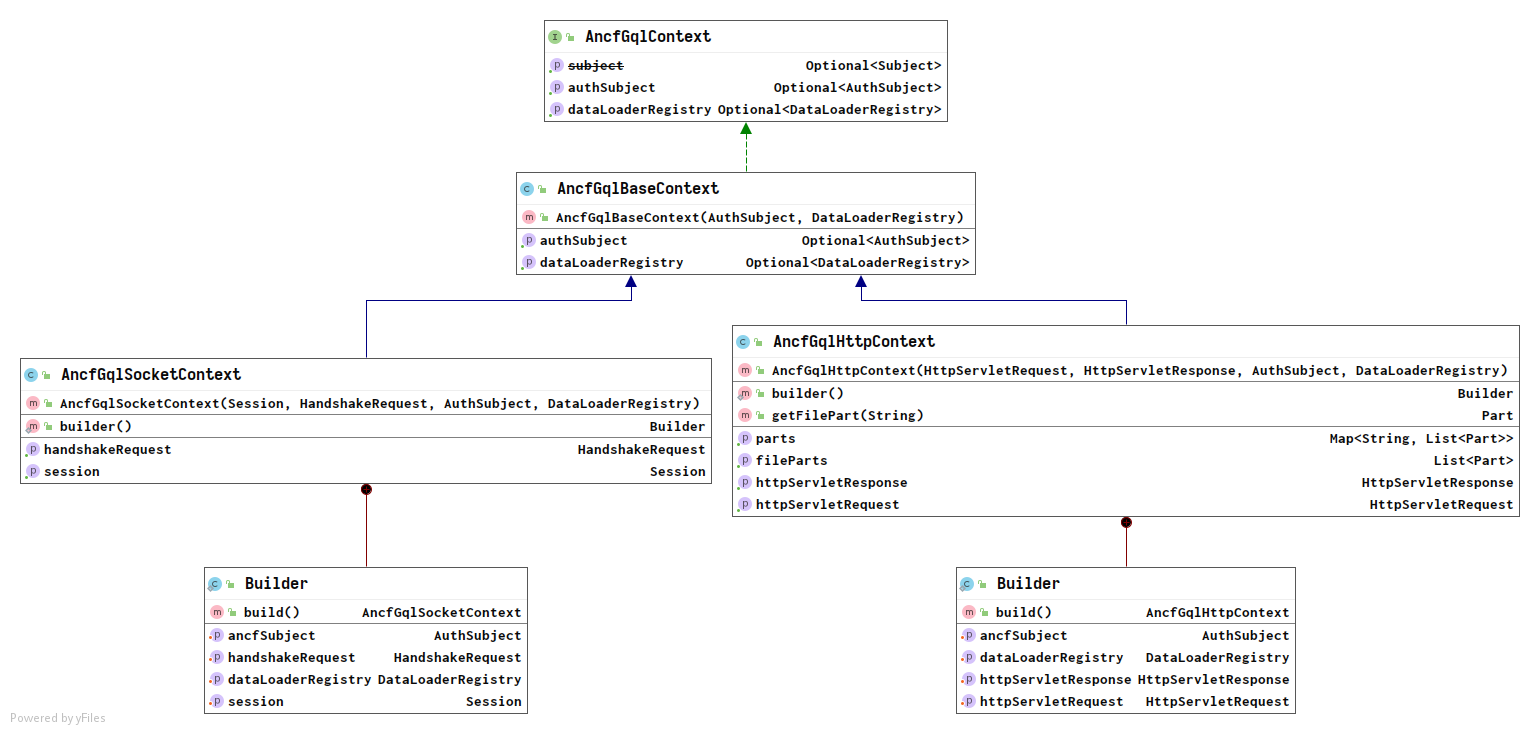
1. ancf-service-oss模块

对象存储服务业务层，使用阿里云对象存储，实现头像的存储，背景图片存储。

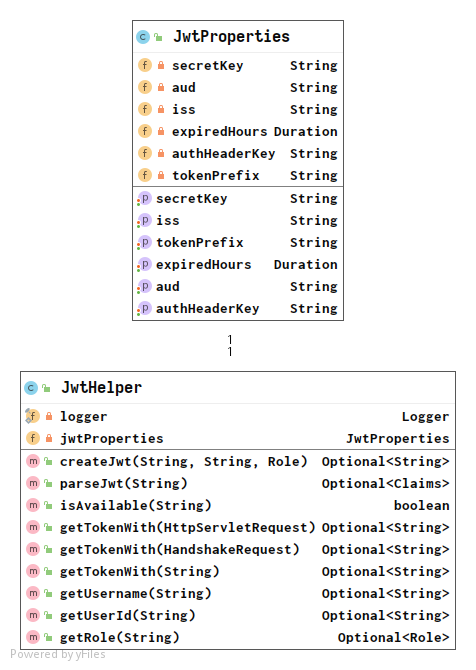


6、ancf-service-auth模块

平台的安全认证实现，重写GraphQL的请求上下文，在其中加入认证对象。在请求进入时，AncfGqlContextBuilder解析token并将解析的已登录用户信息，存储到本模块的本地缓存（ThreadLocal）中，然后初始化AnCFGqlSocketContext或者AnCFGqlHttpContext。

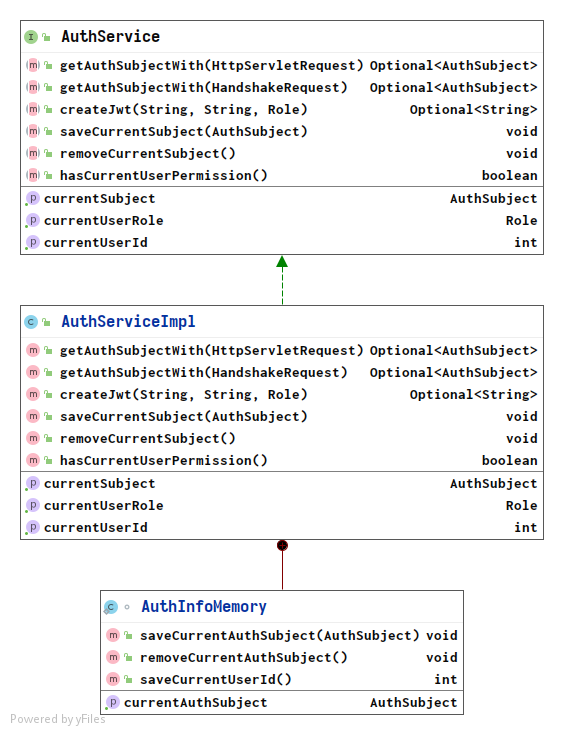
****

[通过@EnableAutoConfigProperties注解，实现JwtProperties相关的属性与application.yaml配置文件绑定。JwtHelper实现一系列jwt](mailto:通过@EnableAutoConfigProperties注解，实现JwtProperties相关的属性与application.yaml配置文件绑定。JwtHelper实现一系列jwt) 生成Token,解析token，验证token，获取用户名，角色信息等等。

****

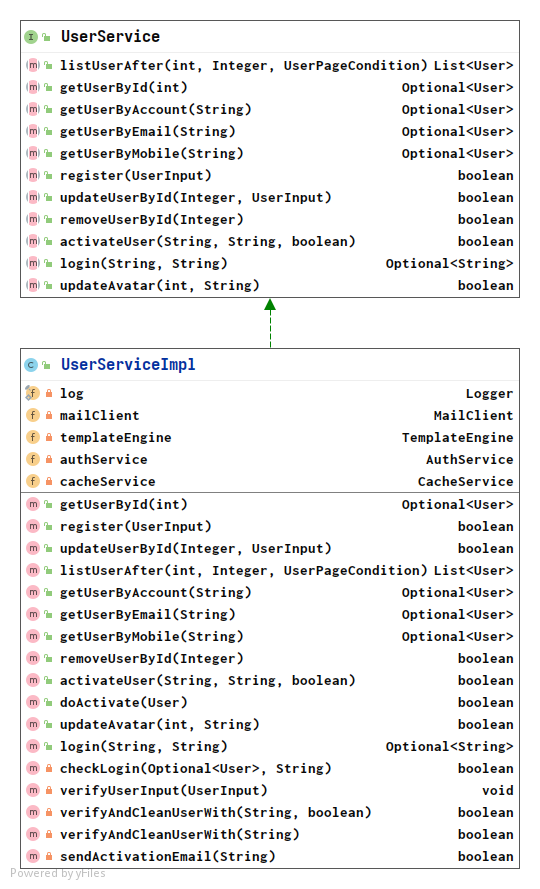
**认证服务接口层**

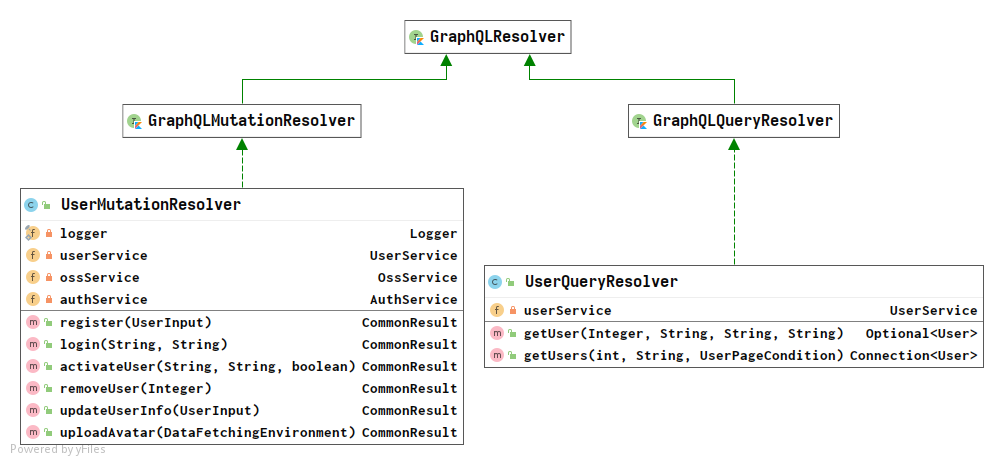
提供了给外部模块访问的接口，创建jwt，解析jwt，获取当前认证用户信息，认证用户角色信息。

****

1. ancf-service-user模块

与用户相关的操作都由该模块完成，用户信息查询，用户注册，用户登录，用户密码修改，用户激活，用户注销，更新头像。

****

****

1. ancf-webapp模块

程序的主启动类，负责配置一些全局配置。

### 4.2.2 部分功能设计与实现

**4.3 平台前端设计与实现**

### 4.3.1 前端技术选取分析

### 4.3.2 页面设计与布局

**4.4 平台前后数据交互策略及实现**

### 4.4.1 交互技术选取技术分析

### 4.4.2 数据通信详细设计

### 4.5.3 数据交互详细实现

# 5 测试与分析

* 1. **测工具介绍**

**PostMan 工具，**

**5.2 测试技术与方法概述**

**5.2 测试用例**

**总结**

**参考文献**